

学生姓名：xxxx

学生学号：xxxx

专业班级：xxxx

指导老师：xxxx

助教：xxxx

2022年12月14日

# 一、问题描述

H大学图书馆

H大学图书馆邀请你建立一个图书馆信息管理系统。请使用面向对象思想完成该问题，具体要求如下：

一、设计一款**文字式交互**的图书管理系统，要求具备注册登录、浏览图书、借还图书等基本功能；

1. 要求以外部文件的形式存储书籍信息、馆藏记录、借阅记录、用户信息等。【可参考提供的书籍清单】

三、图书信息包含但不限于：

* 题名
* ISBN/ISSN
* 作者
* 出版社
* 价格
* ……（可自行扩展）

四、用户基类信息：

* 账户
* 密码
* 角色：1-管理员、2-读者、3-游客（可扩展）

1）管理员权限：

* 读者用户管理：支持对读者的账号进行增、删、改、查；重置读者默认密码为123456；
* 书籍信息管理：支持对馆藏图书进行增、删、改、查；

2）读者模式：

* 读者可以修改自己的密码
* 读者可以搜索图书
* 读者可以借、还图书
* 读者可以查看自己的借阅记录

五、该系统提供根据**任一信息**搜索图书的功能：

* 题名，精确查找到书
* ISBN/ISSN，精确查找到书
* 作者，模糊查找到该作者所有书，字典序排序
* 出版社，模糊查找到该出版社所有书，字典序排序。
* 有多个查询结果显示时，请按字典排序分页显示
* **搜索不考察性能，仅考察功能**

六、该系统提供排行榜功能

* 借阅次数前十的图书
* 借阅次数前十的读者
* 最新出版前十的图书
* ……（可扩展）

七、该系统应有明确的分级菜单显示和跳转功能，可以从上级菜单进入下级菜单，也可以从下级菜单返回上级菜单

八、加分项（总分不超过100分）

（1）图书推荐功能：有过相同借阅记录的书单，则可推荐；

（2）贴近实际的图书馆管理系统，新增未列出的若干功能等；

（3）考虑用户体验，如界面美观、操作方便等；

（4）搜索时性能考察，调查、思考、设计加强搜索性能的方式，此项仅适合学有余力的同学；

（5）其他亮点……

# 二、分析设计

## 数据结构的定义：

1. 输入输出包含在头文件<iostream>中
2. 定义一个类reader，表示每个用户
3. 定义一个类Book，表示每一本书
4. 定义一个类admin,表示管理员
5. 定义一个类RecordReader,记录用户借阅书籍的情况

## 程序主体的设计：

1. 采用分文件编写的方式分别定义不用和类和函数
2. 输出第一级登录界面，输入字符型数，根据输入的字符利用分支分别调用不同的函数
3. 根据选择的用户或管理员分别创建相应的对象
4. 使用对应的vector容器通过流对象将文档中的用户或管理员信息提取，然后对登录信息进行验证
5. 登录验证成功后通过调用成员函数进入下一级操作界面

## 函数设计、参数规格及返回类型：

### Main函数中调用的函数：

1. Menue函数：void型，显示菜单界面
2. Rlogin函数：void型，用于用户登录验证与操作
3. Alogin函数：void型，用于管理员登录验证与操作
4. Vlogin函数：void型，用于游客登录与操作
5. addUser函数：void型，用于注册用户账号

### reader类中：

1. modifyCode函数：void型，用于用户修改密码
2. searchBook函数：void型，用于用户查询图书
3. BorrowAndBack函数：void型，用于用户的借书与还书操作，其中包含borrowBook函数与backBook函数
4. history函数，void型，用于查询用户的借阅历史
5. rank函数：void型，用于用户查看相关排行榜

### admin类中

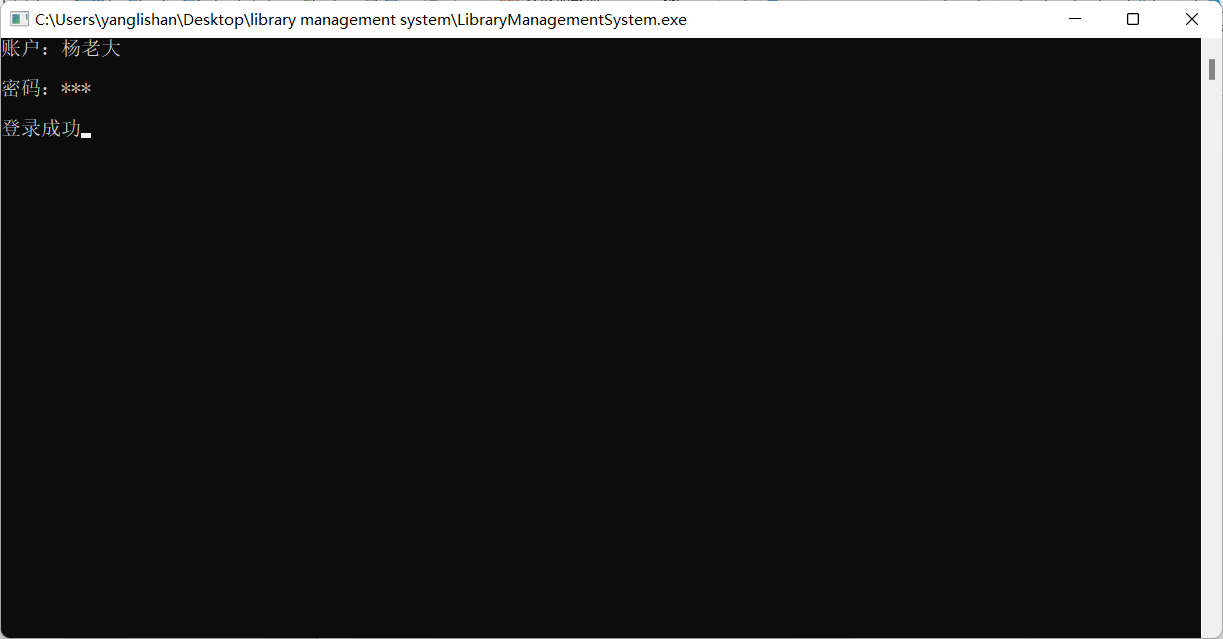
1. manageReader函数：void型，管理员用于管理用户账号，其中包含多个函数
2. resetReader函数：void型，管理员用于重置用户密码
3. checkReader函数：void型，管理员用于查看用户列表
4. addReader函数：void型，管理员用于添加用户账户
5. removeReader函数，void型，管理员用于删除用户账户
6. modifyReader函数：void型，管理员用于修改用户信息
7. addBook函数：void型，管理员用于添加图书
8. modifyBook函数：void型，管理员用于修改图书
9. removeBook函数：void型，管理员用于删除图书
10. checkBook函数：void型，管理员用于查阅书籍
11. modifyCode函数：void型，管理员用来修改自身账户密码

# 三、输入输出测试

## 进入界面

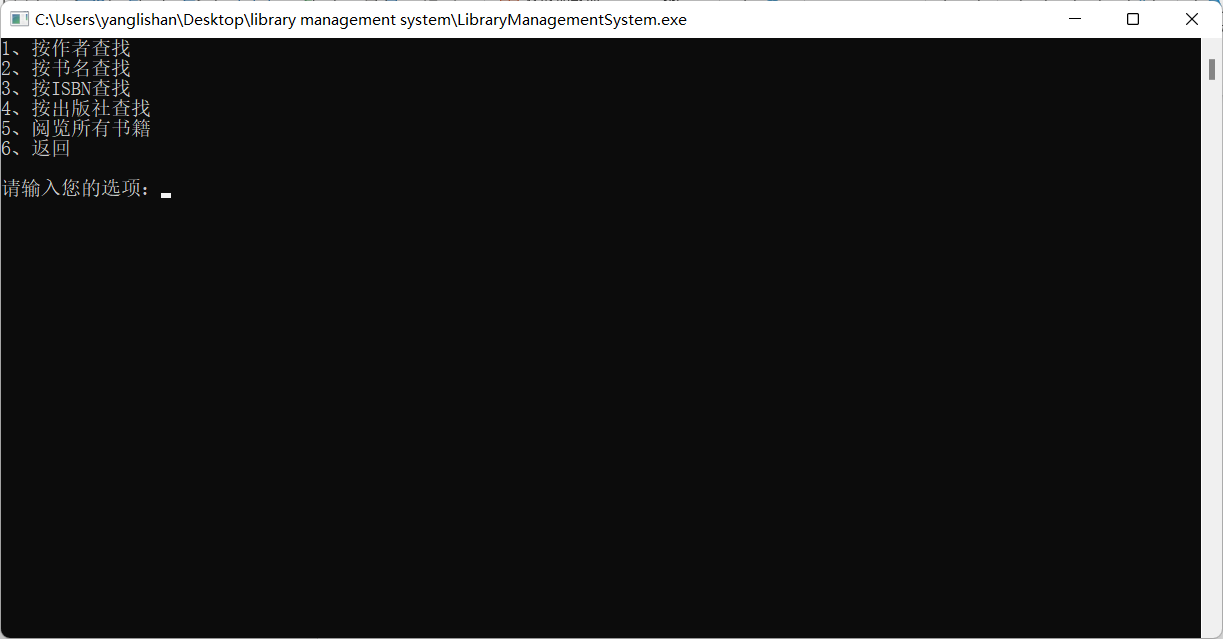


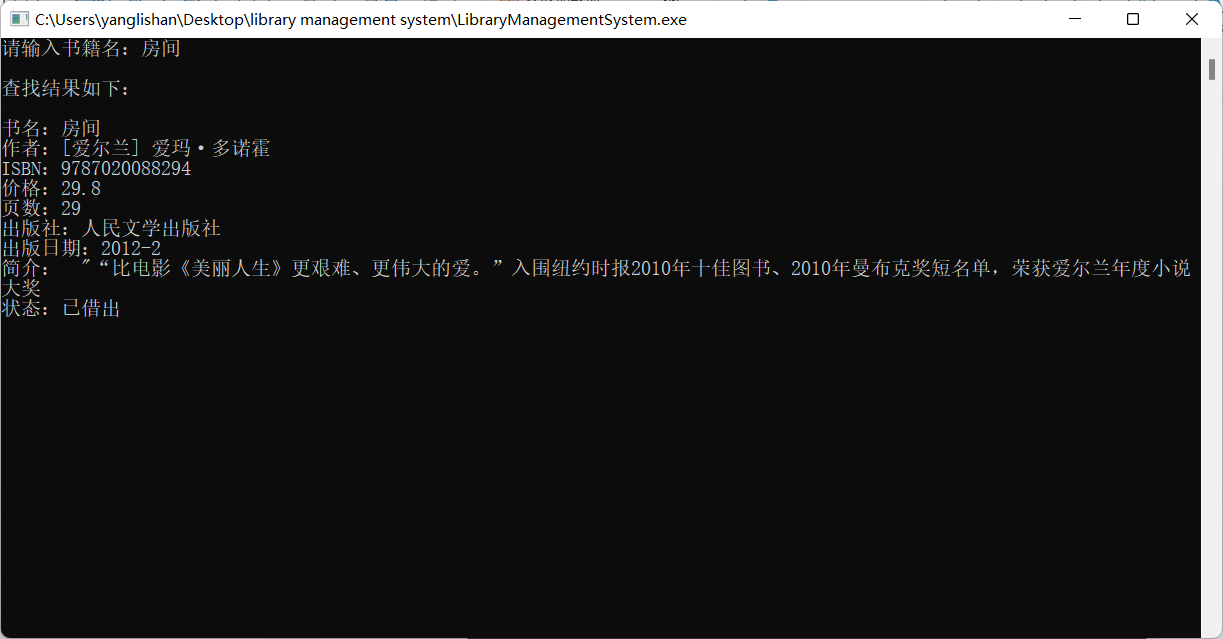
## 用户登录



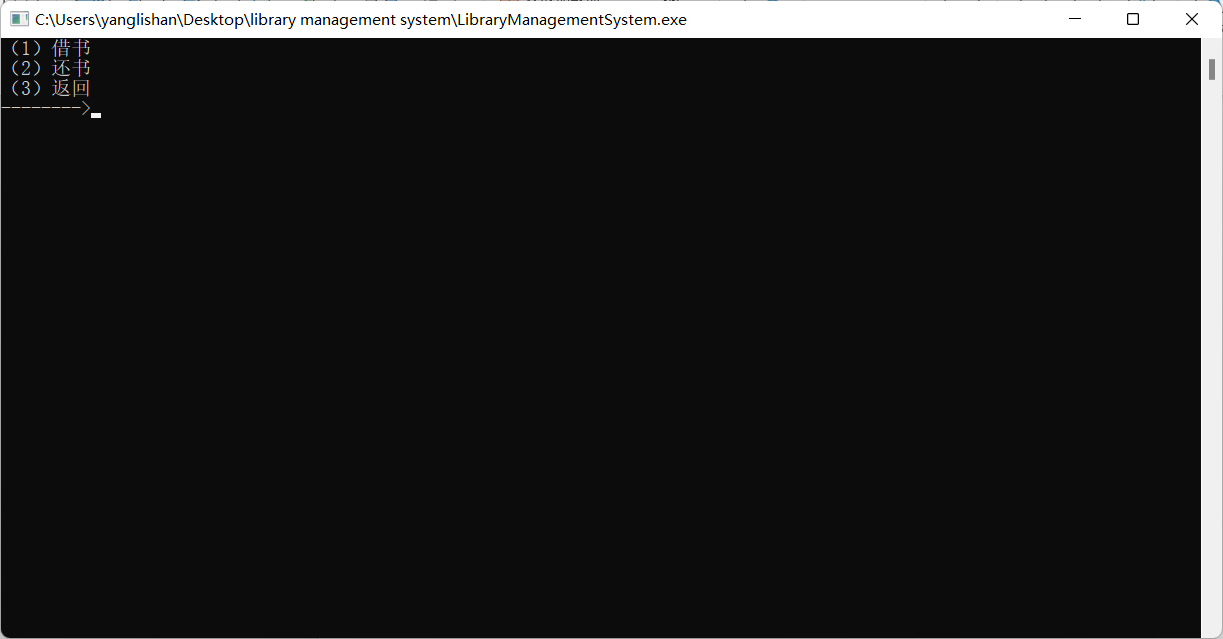


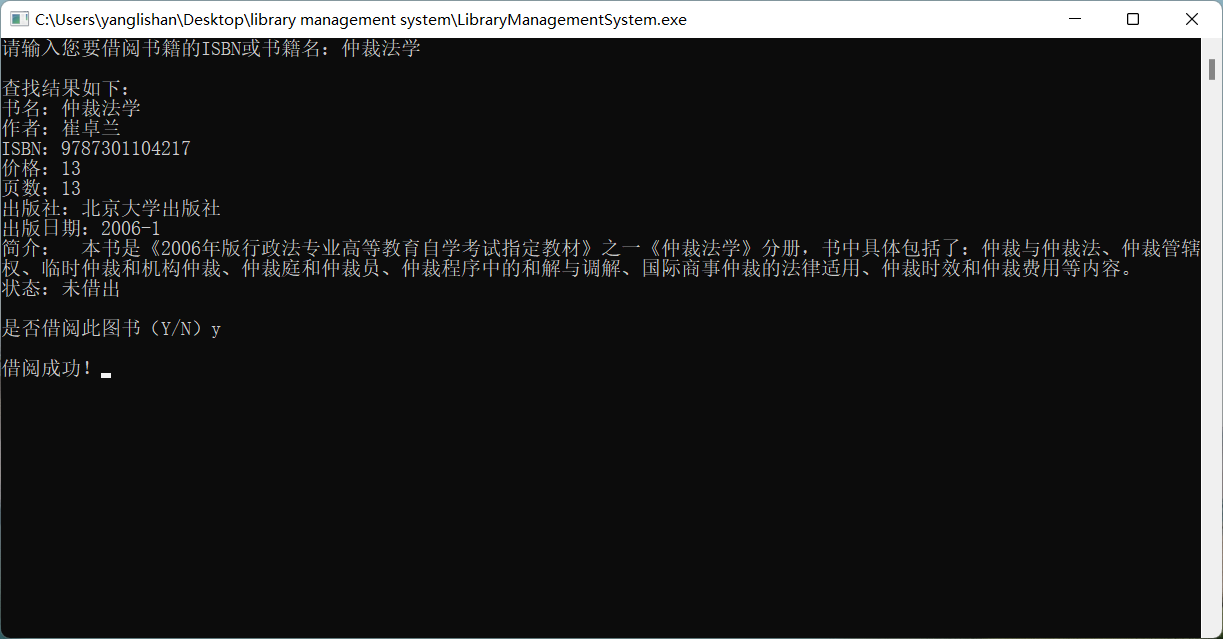
### 搜索书籍

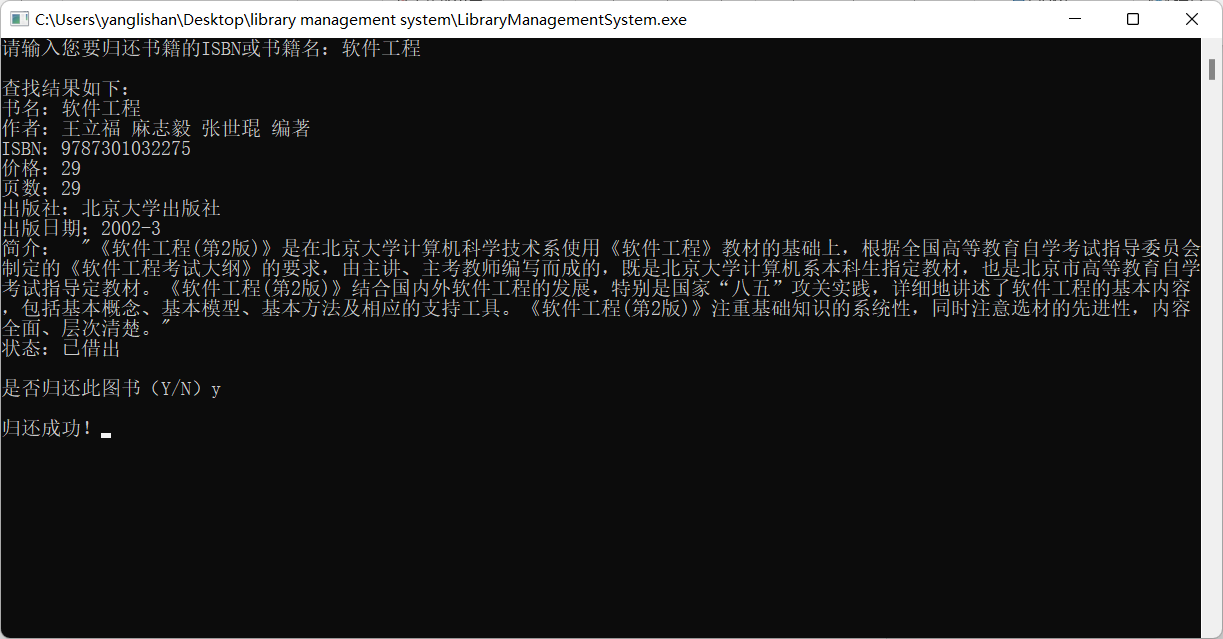




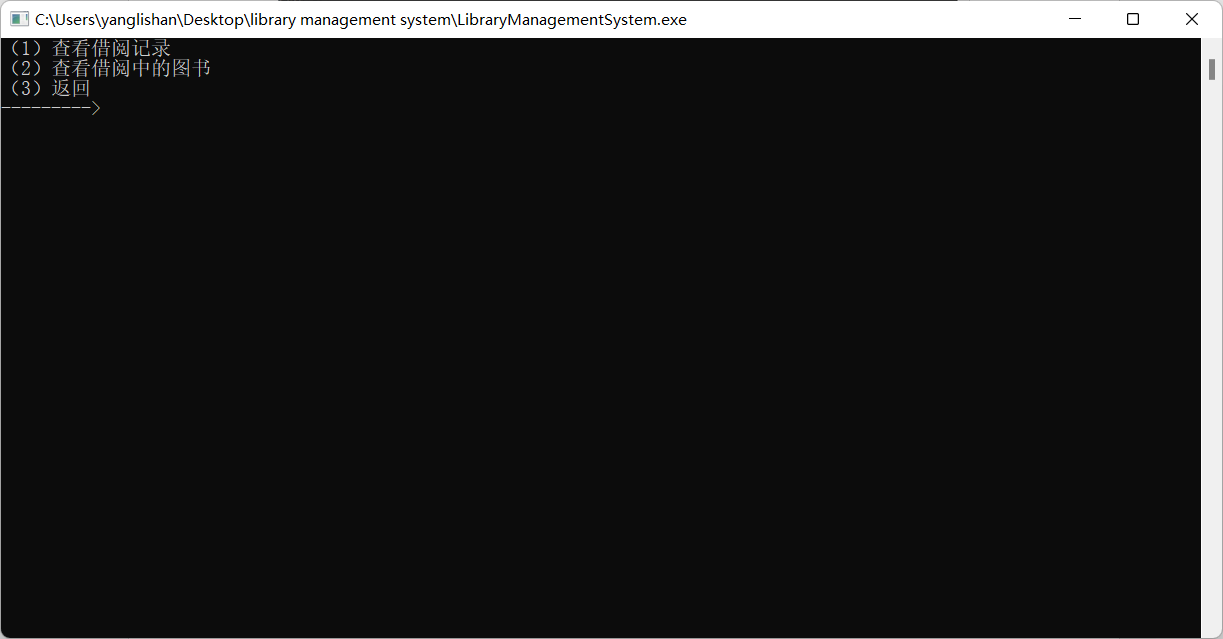
### 借还图书

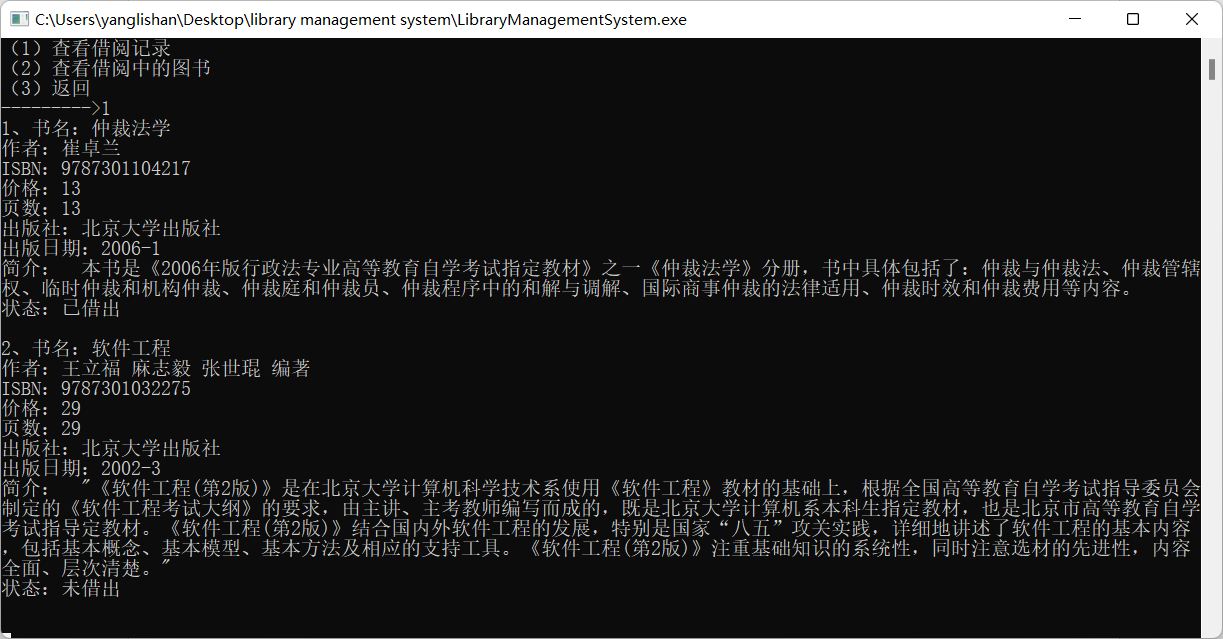


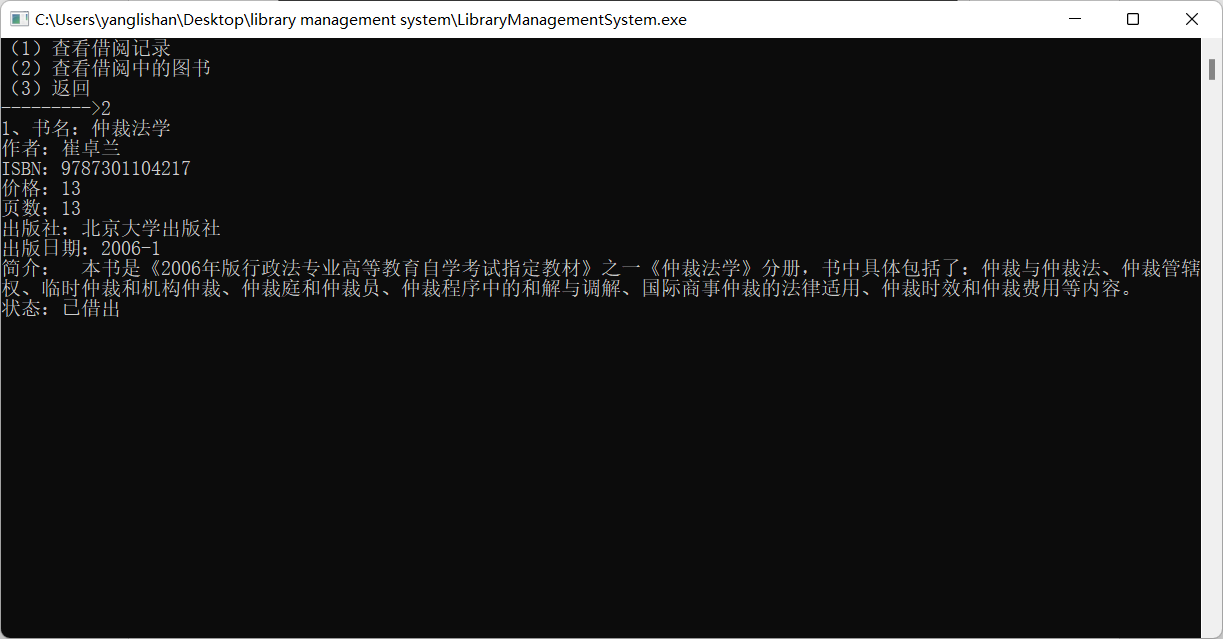




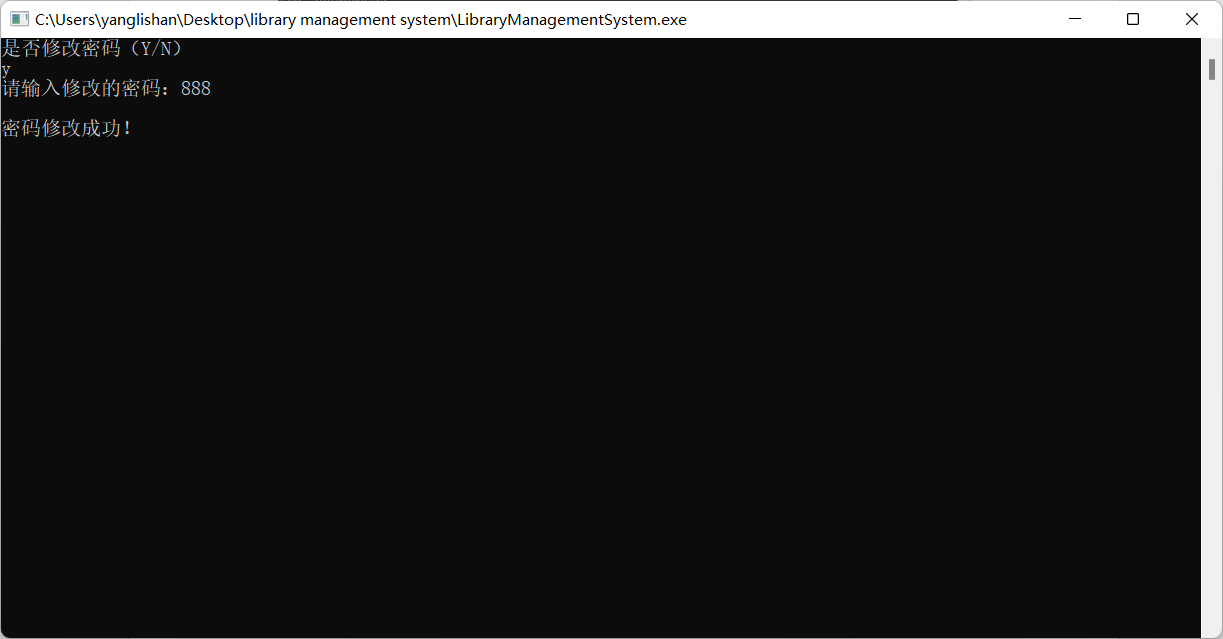
### 借阅记录



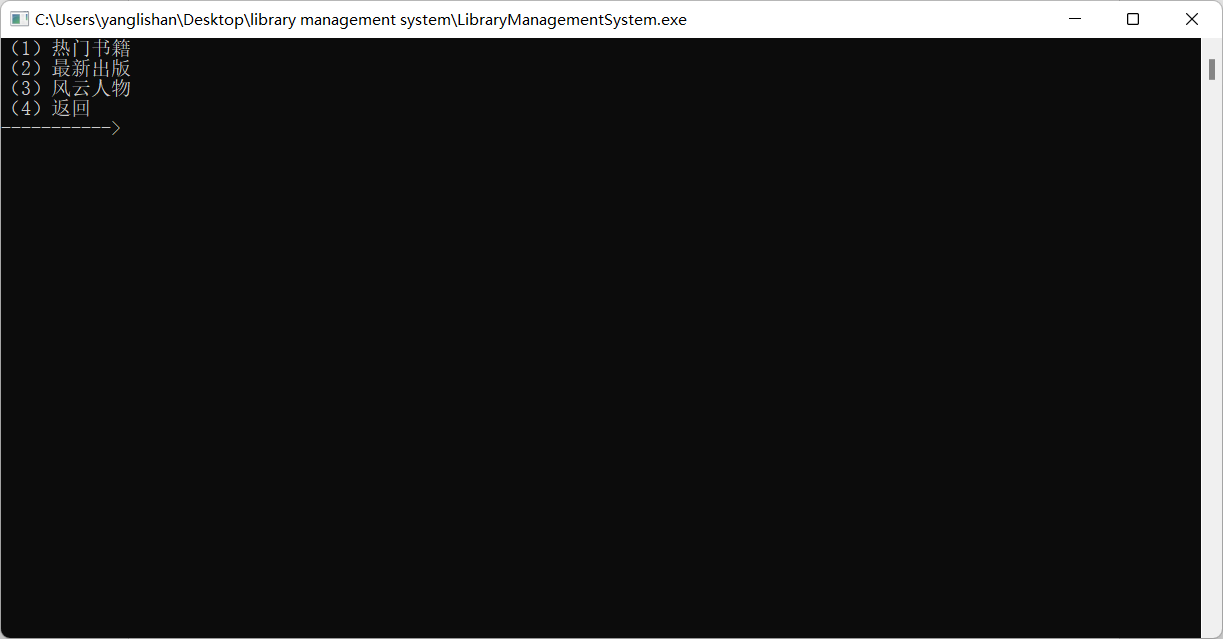


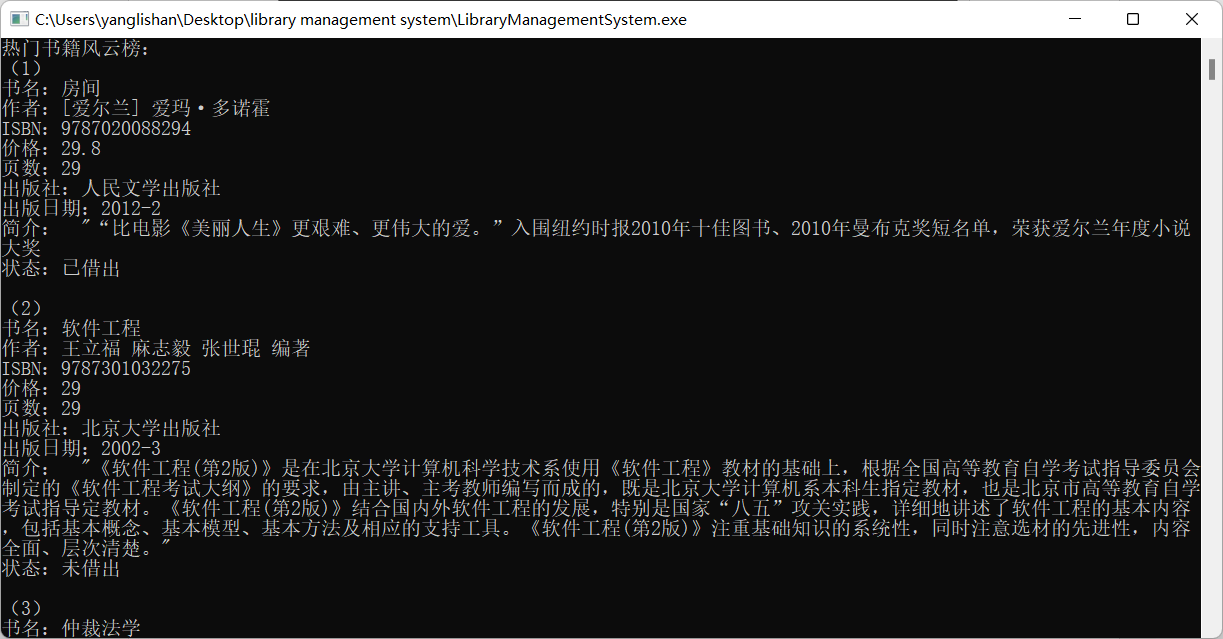


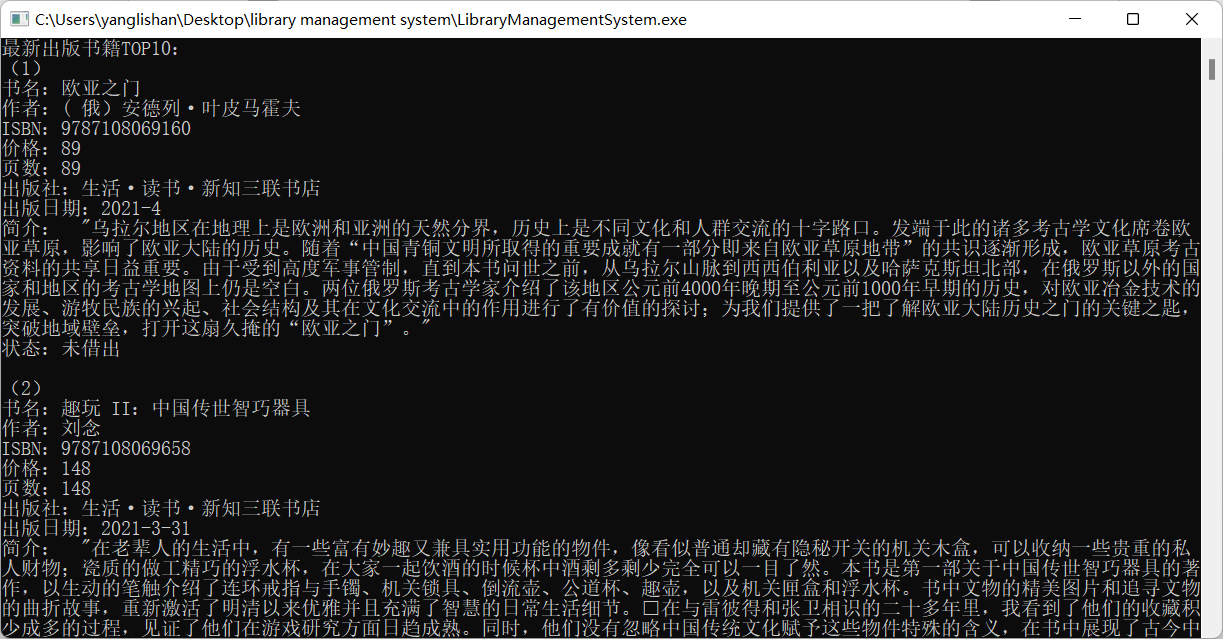
### 修改密码

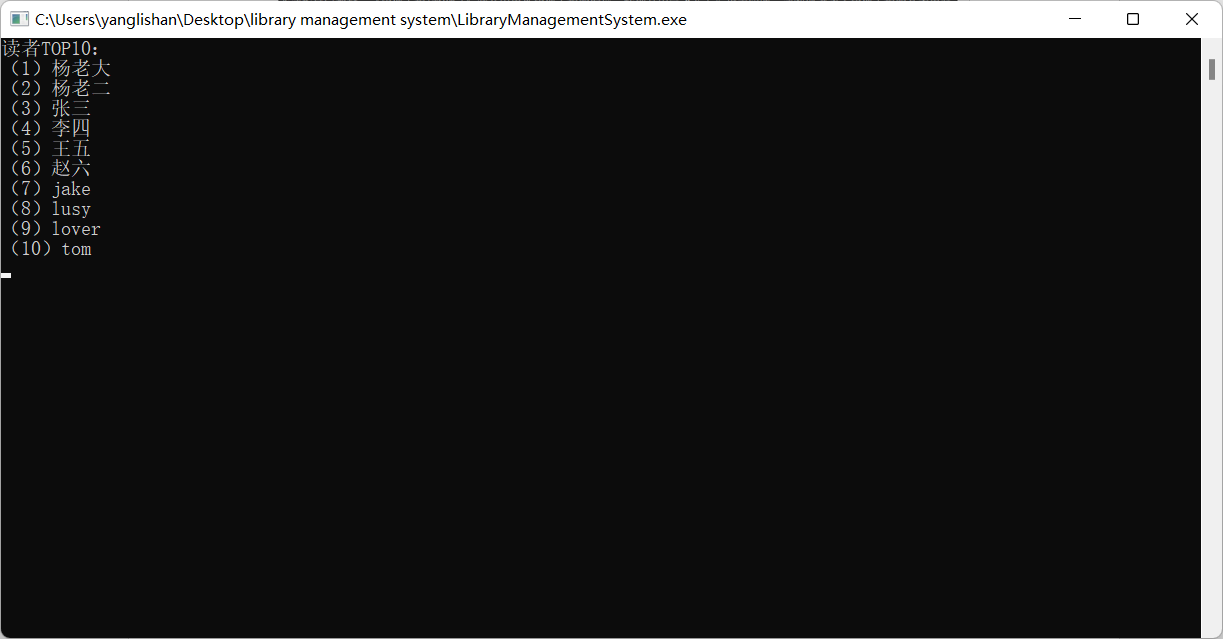


### 查看排行榜





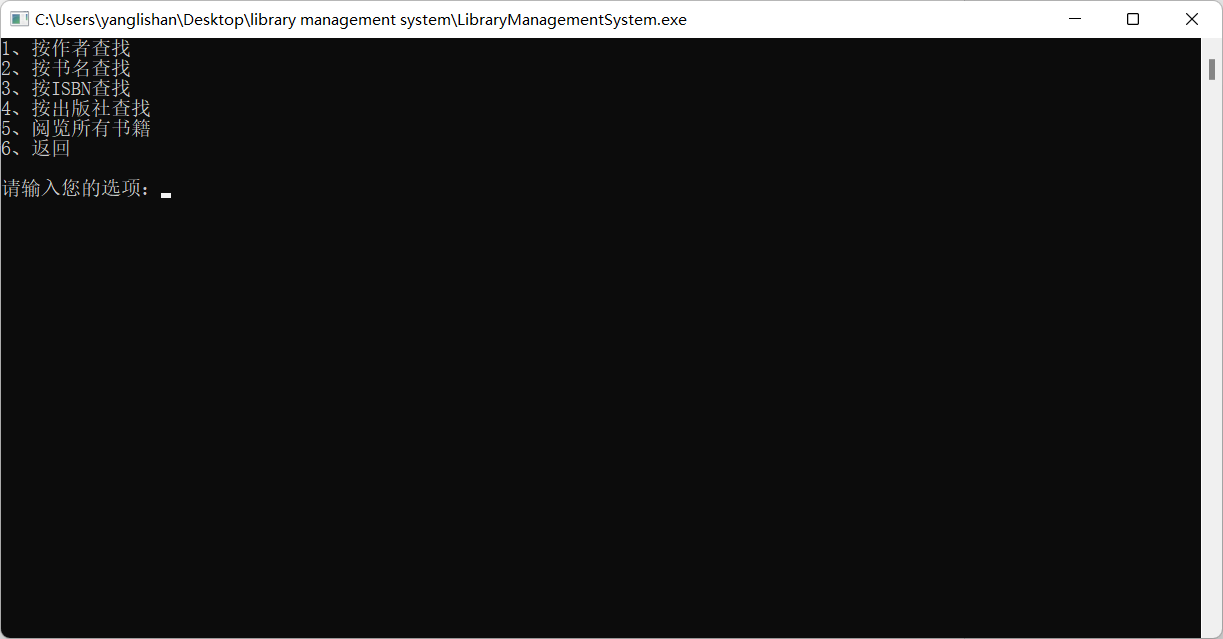




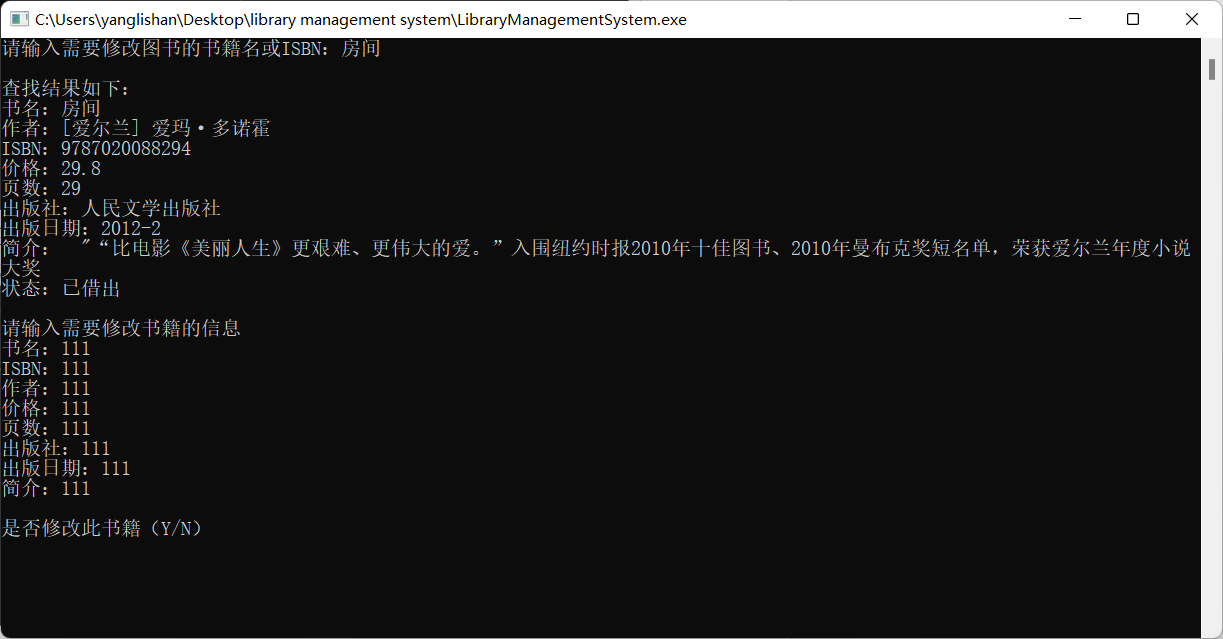
## 管理员界面



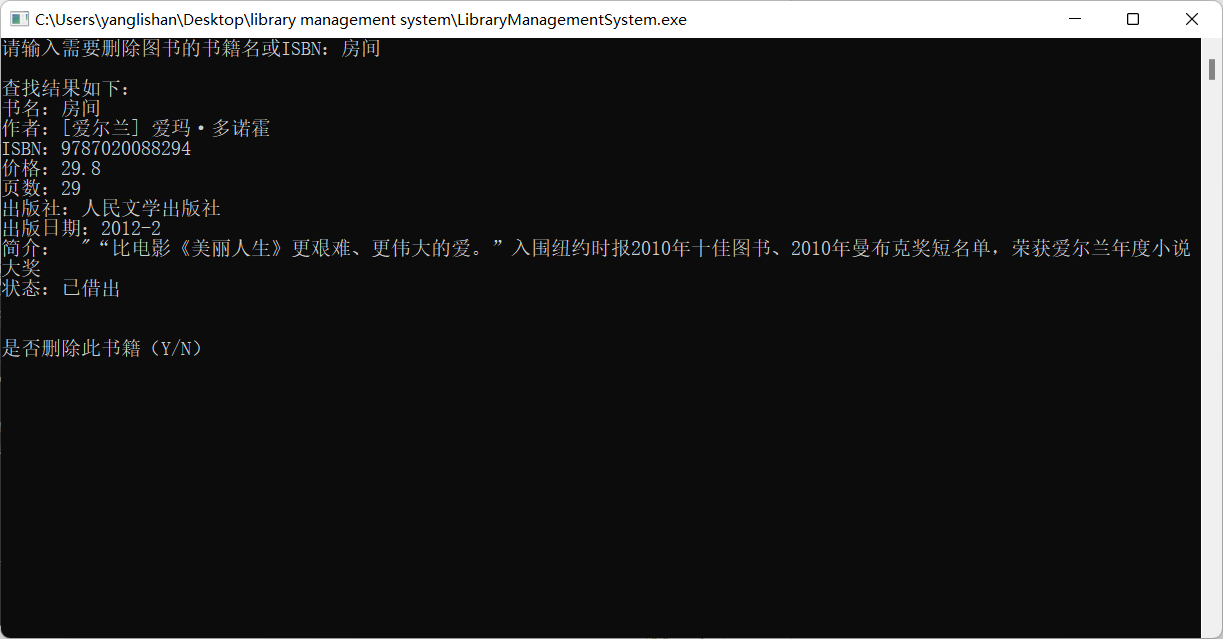
### 查询图书（该功能与读者相似）



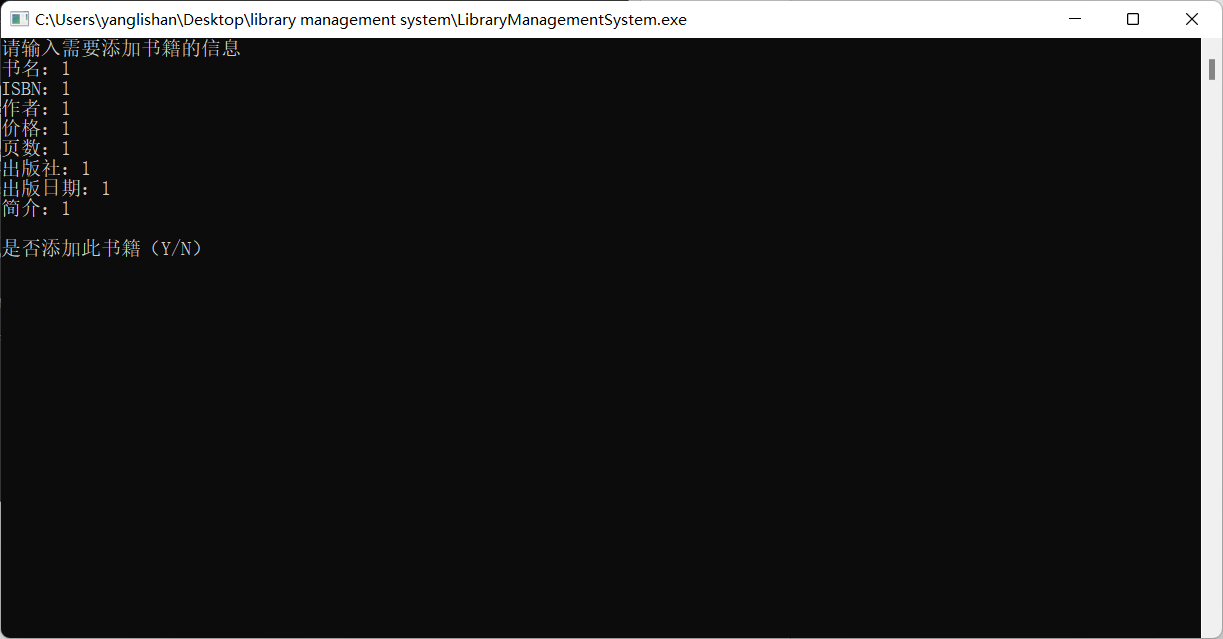
### 修改书籍



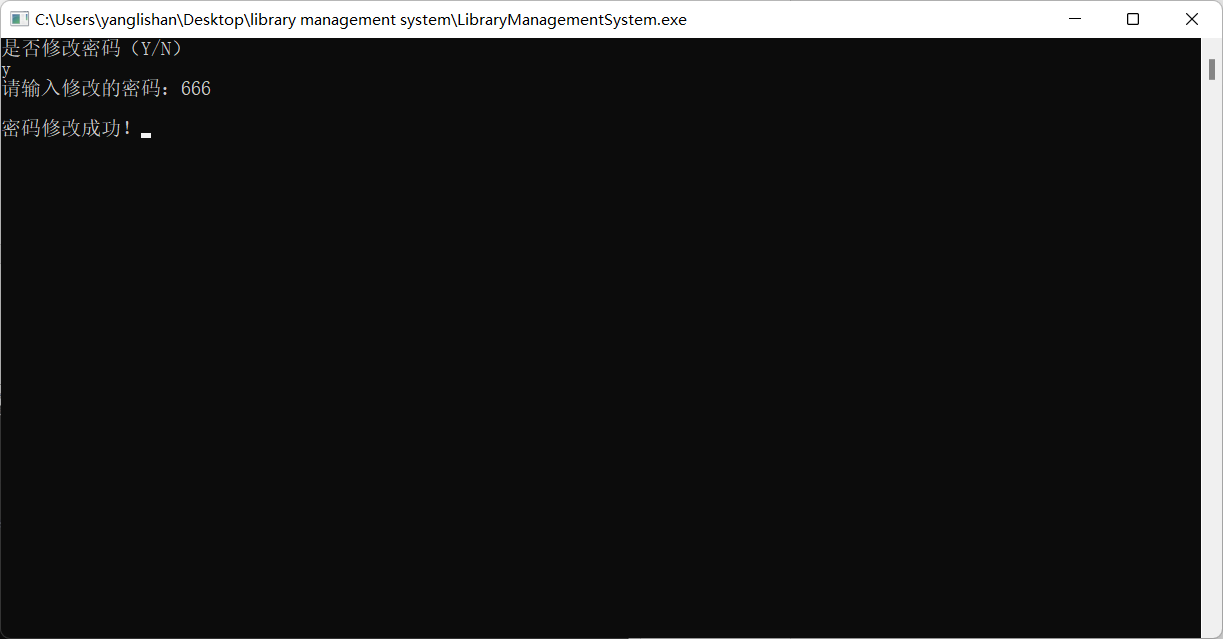
### 删除书籍



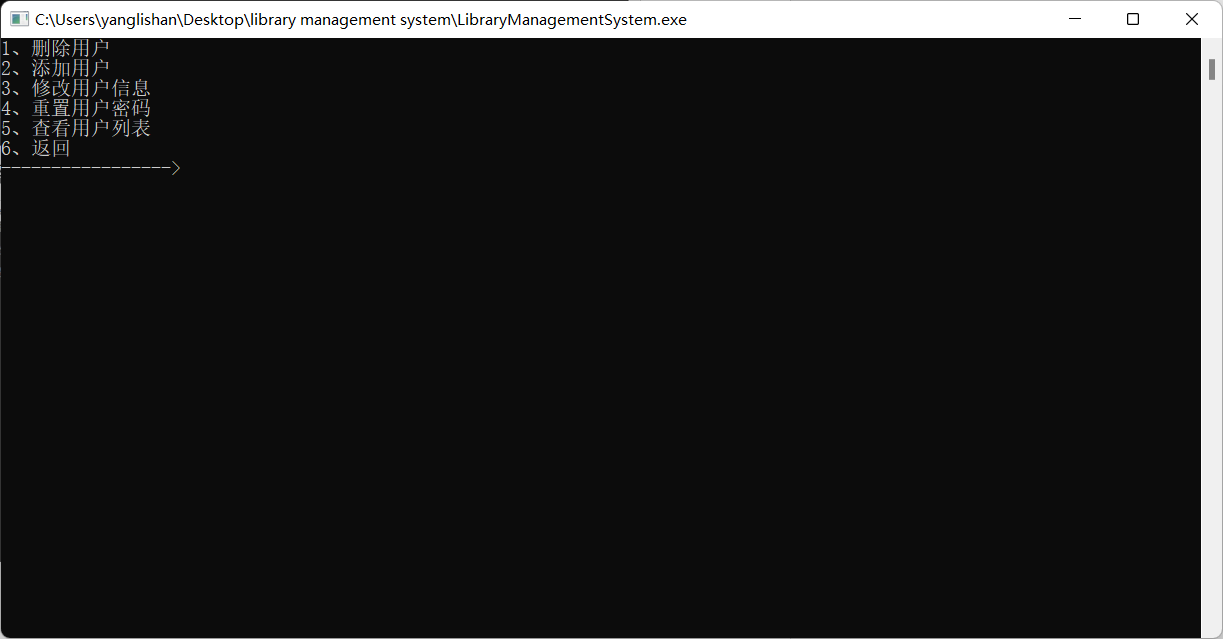
### 添加书籍



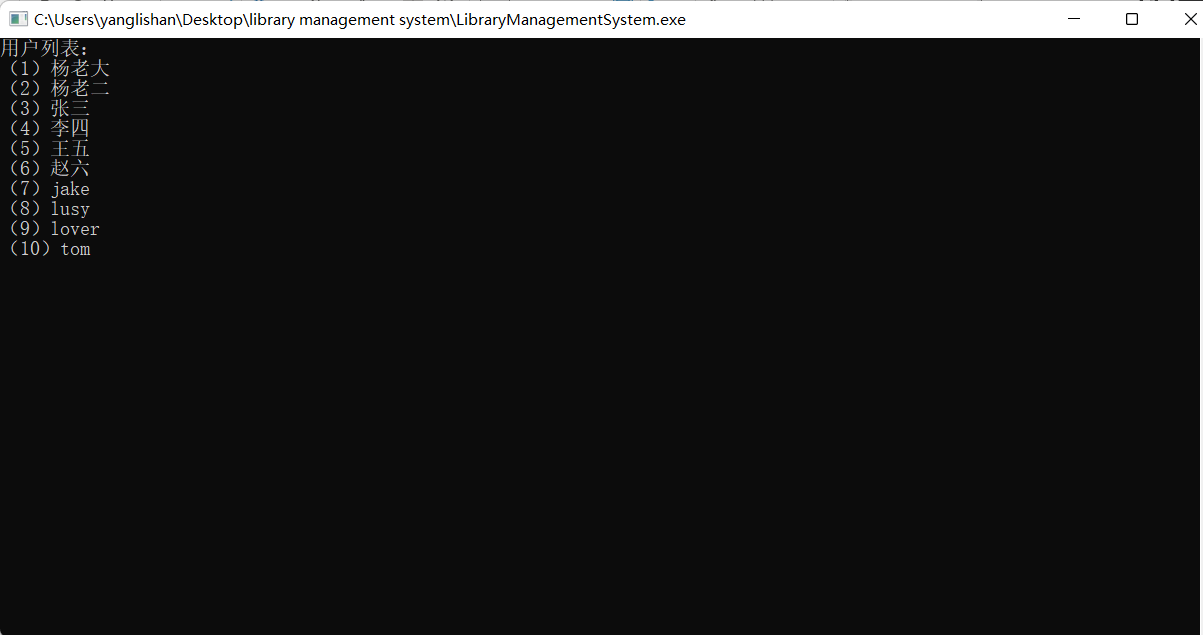
### 修改密码



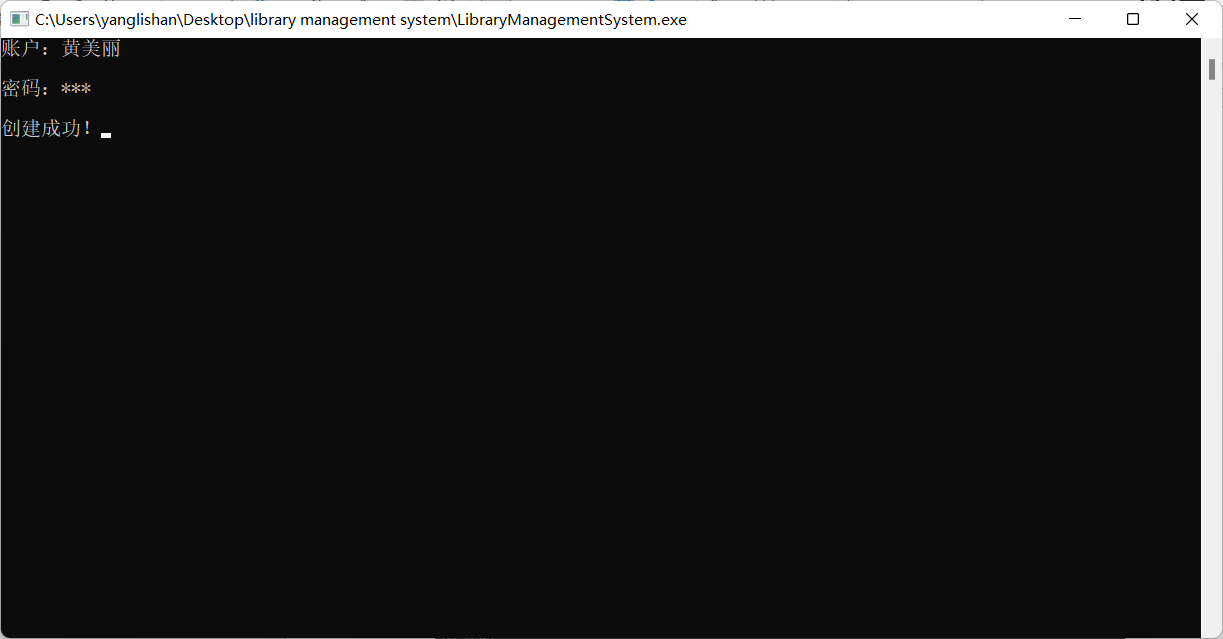
### 读者账户管理



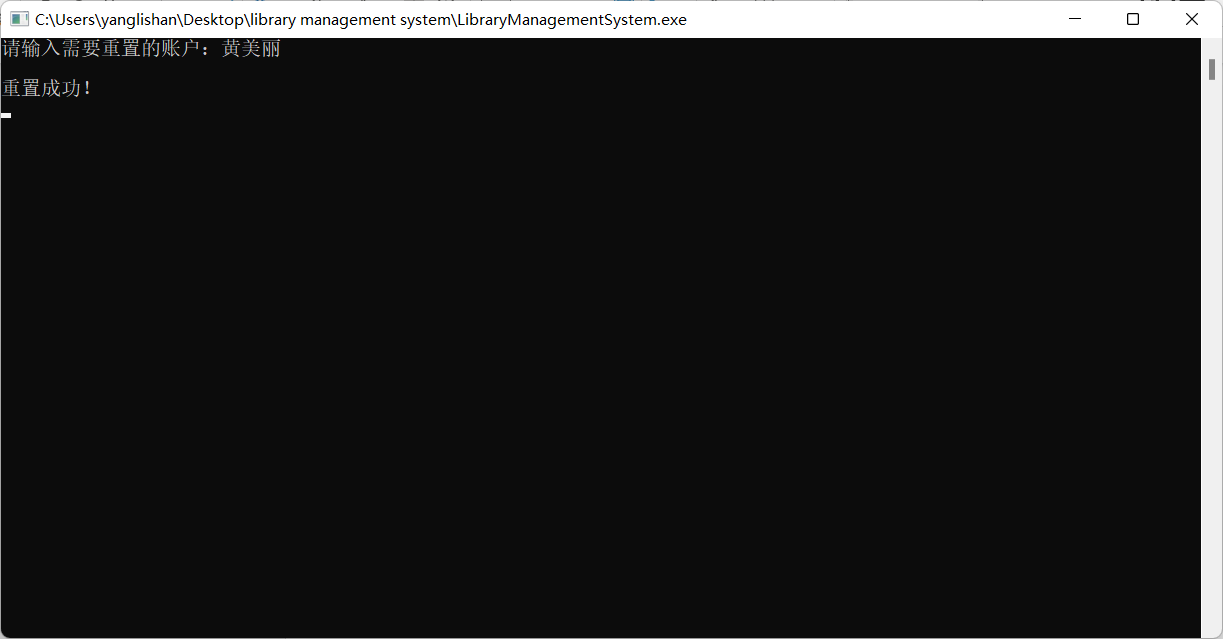
### 用户列表



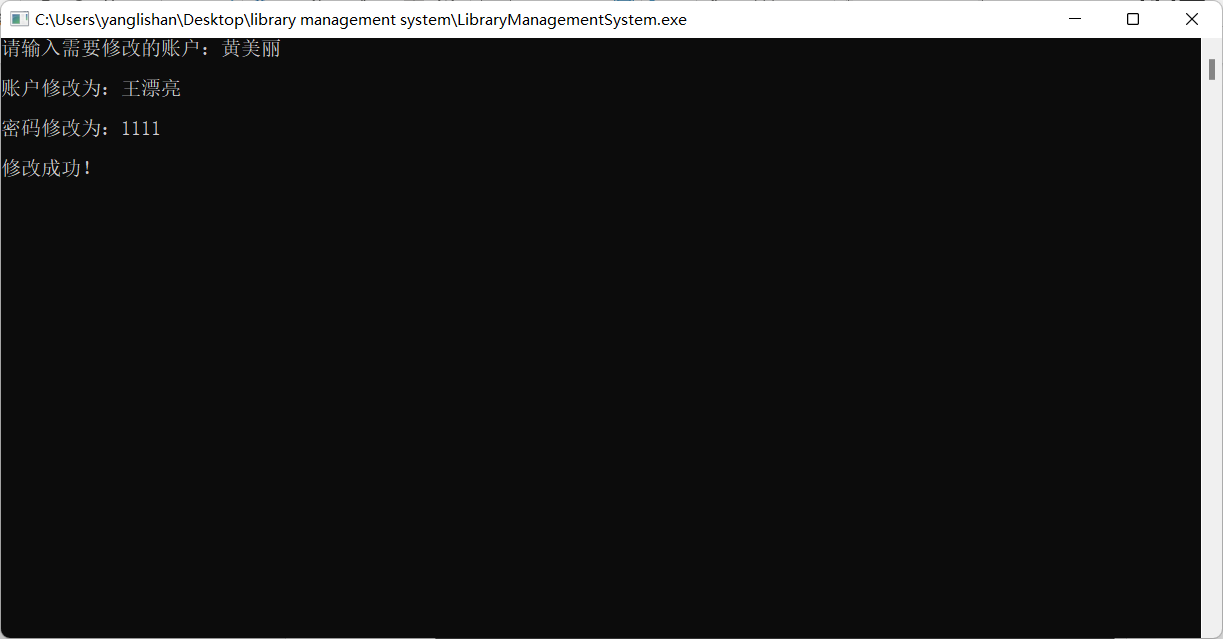
### 添加用户



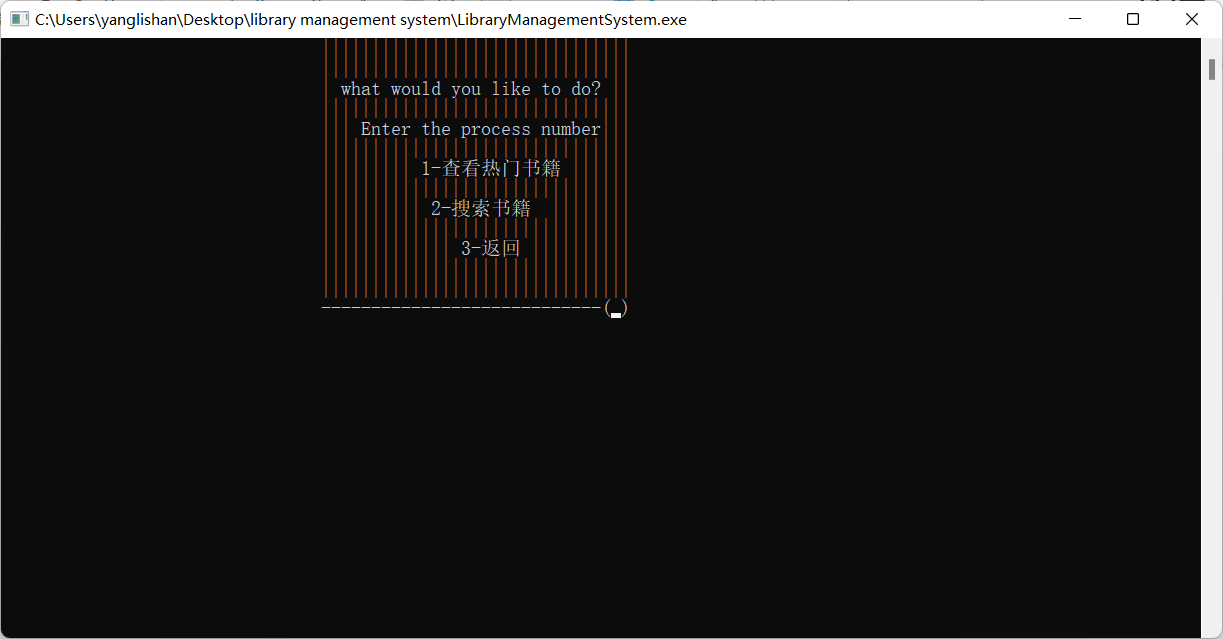
### 重置用户密码为123456



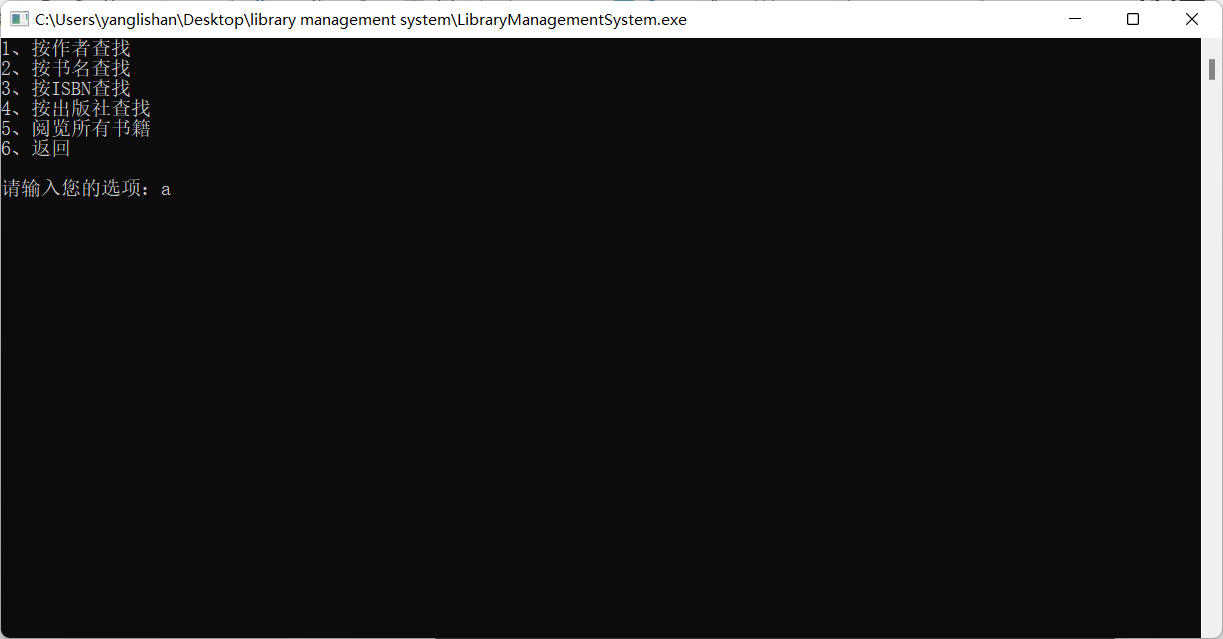
### 修改用户信息



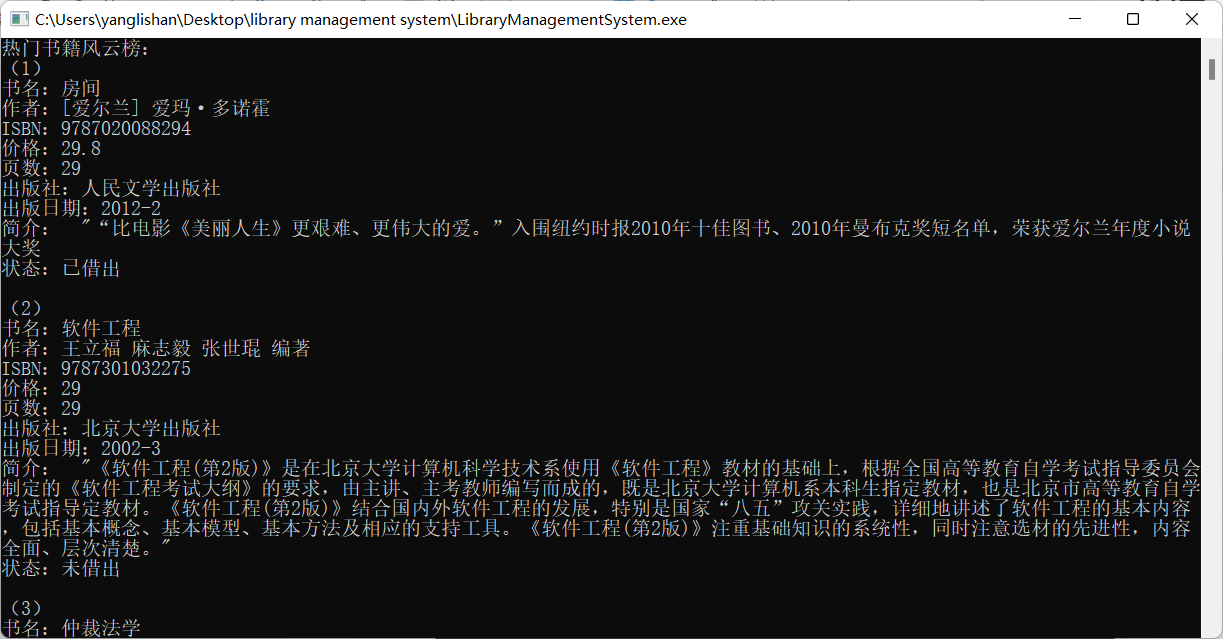
## 游客登录界面



### 搜索书籍（功能与一般用户相同）



### 查看热门书籍



# 四、分析与总结

## 1.实验分析

* 1. 定义Book、reader、admin、RecorReader类
  2. 在类中写必要的成员函数
  3. 在主函数中根据主界面的选择调用不同函数，函数中包含不同对象的创建
  4. 调用类中的成员函数实现各种功能

## 2遇到的问题:

1. 在分文件中的.cpp定义vector容器声明失败
2. Ifstream类进行文件读入时，在>> 与getline之间后者会录入空格

## 3.解决的方法:

1. 可将vector定义在main函数中作为全局的容器，在分文件的.cpp文件中通过extern关键字声明
2. 在>>与getline之间通过.get()忽略空格影响

## 4学习到的新知识:

通过参考github上的相关代码以及网络资料和老师课上的讲解,结合在编译器上的多次实践尝试,学习了面向对象的有关知识,如何运用类及对象,使用函数调用类内成员, 各个函数的使用及,如何清屏等知识.